

(12) PATENT
(19) AUSTRALIAN PATENT OFFICE

(11) Application No. AU 200071933 B2
(10) Patent No. 748978

(54) Title
Transforming multi-dimensional data

(51)⁷ International Patent Classification(s)
G06T 009/00 G06F 017/14

(21) Application No: **200071933**

(22) Application Date: **2000.11.30**

(30) Priority Data

(31) Number	(32) Date	(33) Country
PQ4573	1999.12.10	AU
PQ4573	1999.12.10	AU

(43) Publication Date : **2001.06.14**

(43) Publication Journal Date : **2001.06.14**

(44) Accepted Journal Date : **2002.06.13**

(71) Applicant(s)
Canon Kabushiki Kaisha

(72) Inventor(s)
Timothy John Lindquist

(74) Agent/Attorney
SPRUSON and FERGUSON,GPO Box 3898,SYDNEY NSW 2001

(56) Related Art
EP 836322

BEST AVAILABLE COPY

Abstract

TRANSFORMING MULTI-DIMENSIONAL DATA

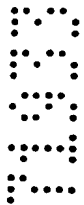
5

The method transforms the coefficients of a two-dimensional data set. The method comprises a scanning step (804), where the coefficients are scanned in raster scan order. The method then vertically convolves (806) a number of said coefficients to produce a corresponding intermediate transformed coefficient. The method then stores (808) the

10 corresponding intermediate transformed coefficient in a finite memory array. The current corresponding intermediate coefficient is stored at a start of the array and previously said stored intermediate coefficients are each shifted once down the array with a previously said stored intermediate coefficient at an end of the array being shifted out of the array. The method then convolves (810) the intermediate transformed coefficients currently

15 stored in said finite memory array to produce a corresponding transformed coefficient.

Fig. 8



AUSTRALIA
PATENTS ACT 1990
COMPLETE SPECIFICATION

FOR A STANDARD PATENT

ORIGINAL

Name and Address
of Applicant :

Canon Kabushiki Kaisha
30-2, Shimomaruko 3-chome, Ohta-ku
Tokyo 146
Japan

Actual Inventor(s):

Timothy John Lindquist

Address for Service:

Spruson & Ferguson
St Martins Tower
31 Market Street
Sydney NSW 2000

Invention Title:

Transforming Multi-Dimensional Data

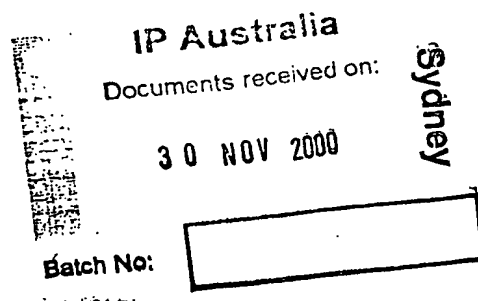
ASSOCIATED PROVISIONAL APPLICATION DETAILS

[33] Country
AU

[31] Applic. No(s)
PQ4573

[32] Application Date
10 Dec 1999

The following statement is a full description of this invention, including the best method of performing it known to me/us:-



TRANSFORMING MULTI-DIMENSIONAL DATA

Technical Field of the Invention

The present invention relates generally to transforming a multi-dimensional data set in a first domain to a multi-dimensional data set in a second domain and, in particular, to transforming image data using a separable two-dimensional Discrete Wavelet Transform (DWT).

Background

The field of digital data compression and in particular digital image compression has attracted great interest for some time.

10 In the field of digital image compression, many different techniques have been utilised. In particular, one popular technique is the JPEG standard, which utilises the discrete cosine transform to transform standard size blocks of an image into corresponding cosine components. In this respect, the higher frequency cosine components are heavily quantised so as to assist in obtaining substantial compression.

15 The heavy quantisation is an example of a lossy technique of image compression. The JPEG standard also provides for the subsequent lossless compression of the transform coefficients.

20 Recently, the field of wavelet transforms has gained great attention as an alternative form of data compression. The wavelet transform has been found to be highly suitable in representing data having discontinuities such as sharp edges. Such discontinuities are often present in image data or the like.

Typically, data compression using wavelet techniques is a two step process. It comprises, firstly, a transform phase, during which the wavelet transform of the data set is calculated, and secondly a subsequent coding stage during which the resultant data set

25 from the transform operation is separated into segments which are then coded using a specific coder. In decompression, the reverse occurs, with coded blocks being first decoded, and subsequently the inverse wavelet transform being applied to generate the final decompressed output.

30 The inverse discrete wavelet transform is computationally intensive, and traditional decoders resort to large memory stores to store intermediate results generated during the inverse operation through the separate applications of line based horizontal and vertical filters. These large memory stores adversely affect the performance of the decoders.

Summary of the Invention

It is an object of the present invention to substantially overcome, or at least ameliorate, one or more disadvantages of existing arrangements.

According to one aspect of the invention, there is provided a method of
5 transforming a two-dimensional data set in a first domain to a two dimensional data set in
a second domain, wherein the two-dimensional data set of the first domain comprises
coefficients arranged in rows and columns, and the method performs the following steps
for each said coefficient in sequence in a first direction: convolving a number of said
coefficients that are arranged in a second direction transverse to the first direction to
10 produce a corresponding intermediate transformed coefficient; storing said corresponding
intermediate transformed coefficient in a finite memory array, wherein said corresponding
intermediate coefficient is stored at a start of the array and previously said stored
intermediate coefficients are each shifted once down the array with a previously said
stored intermediate coefficient at an end of the array being shifted out of the array; and
15 convolving said intermediate transformed coefficients currently stored in said finite
memory array to produce a corresponding transformed coefficient.

According to another aspect of the invention, there is provided apparatus for
transforming a two-dimensional data set in a first domain to a two dimensional data set in
a second domain, wherein the two-dimensional data set of the first domain comprises
20 coefficients arranged in rows and columns, and the apparatus comprises: a first convolver
for convolving a number of said coefficients to produce a corresponding intermediate
transformed coefficient; a finite memory array for storing said corresponding intermediate
transformed coefficient, wherein said corresponding intermediate coefficient is stored at a
start of the array and previously said stored intermediate coefficients are each shifted once
25 down the array with a previously said stored intermediate coefficient at an end of the
array being shifted out of the array; and a second convolver for convolving said
intermediate transformed coefficients currently stored in said finite memory array to
produce a corresponding transformed coefficient.

According to still another aspect of the invention, there is provided computer
30 readable medium including a computer program for transforming a two-dimensional data
set in a first domain to a two dimensional data set in a second domain, wherein the two-
dimensional data set of the first domain comprises coefficients arranged in rows and
columns, and the computer program comprising: code for convolving a number of said
coefficients to produce a corresponding intermediate transformed coefficient; code for

storing said corresponding intermediate transformed coefficient in a finite memory array, wherein said corresponding intermediate coefficient is stored at a start of the array and previously said stored intermediate coefficients are each shifted once down the array with a previously said stored intermediate coefficient at an end of the array being shifted out of the array; and code for convolving said intermediate transformed coefficients currently stored in said finite memory array to produce a corresponding transformed coefficient.

According to still another aspect of the invention, there is provided an inverse two-dimensional separable wavelet transformer comprising a plurality of stages, at least one said stage comprising: a first filter for inverse one-dimensional wavelet transforming in a first direction coefficients to produce corresponding intermediate transformed coefficients of a first type; an adder for adding said corresponding intermediate transformed coefficients of a first type to respective intermediate transformed coefficients of a first type from another stage to produce corresponding intermediate transformed coefficients of a second type; a second filter for inverse one-dimensional wavelet transforming in a second direction transverse to the said first direction said corresponding intermediate coefficients of a second type to produce corresponding transformed coefficients, the second filter comprising: a shift register coupled to the adder for storing said corresponding intermediate transformed coefficients of a second type, and an arrangement for convolving said intermediate transformed coefficients of the second type currently stored in said shift register to produce a corresponding transformed coefficient.

Brief Description of the Drawings

A number of preferred arrangements of the present invention will now be described with reference to the drawings, in which:

Fig. 1 is a schematic overview of a prior art two-dimensional separable wavelet transformer;

Fig. 2 is a schematic overview of a prior art inverse two-dimensional separable wavelet transformer;

Fig. 3 is a schematic representation of part of an inverse two-dimensional separable DWT transformer in accordance with a first embodiment;

Fig. 4 is a schematic block diagram of a parallel convolver suitable for use as the vertical filter 302 shown in Fig. 3;

Fig. 5 is a schematic block diagram of a sequential convolver suitable for use as the horizontal filter 304 shown in Fig. 3;

Fig. 6 is a schematic block diagram of another sequential convolver suitable for use as the horizontal filter 304 shown in Fig. 3;

Fig. 7 is a schematic representation of part of an inverse two-dimensional separable DWT transformer in accordance with a second embodiment;

5 Fig. 8 is a flow diagram of a method of transforming a two dimensional data set in a first domain to a two-dimensional data set in a second domain in accordance with a third embodiment; and

Fig. 9 is a schematic block diagram of a general-purpose computer upon which the third embodiment can be practiced.

10 **Detailed Description including Best Mode**

Where reference is made in any one or more of the accompanying drawings to steps and/or features, which have the same reference numerals, those steps and/or features have for the purposes of this description the same function(s) or operation(s), unless the contrary intention appears.

15 The principles of the preferred embodiments described herein have general applicability to the implementation of separable multi-dimensional convolution kernels and in particular symmetric separable two dimensional convolution kernels. However, for ease of explanation, the first embodiment has been described with reference to a separable two-dimensional inverse discrete wavelet transform (DWT) of transformed
20 image data. Although the first embodiment is described with reference to the decompression of image data, it will be readily evident that the invention is not limited thereto. For examples of the many different applications of wavelet analysis to signals, reference is made to a survey article entitled "Wavelet Analysis" by Bruce et. al. appearing in IEEE spectrum, October 1996 pages 26 to 25. For a discussion of the
25 different applications of wavelets in computer graphics, reference is made to "Wavelets for Computer Graphics", page 5, I. Stollnitz et. al. published 1996 by Morgan Kaufmann Publishers, Inc.

Turning now to Fig. 1, there is shown a schematic overview of a prior art two-dimensional separable wavelet transformer 100. As shown, a digital input image 102 is
30 fed to an input 104 of the wavelet transformer 100. The digital image 102 comprises pixels arranged in a plurality of rows in the horizontal direction and a plurality of columns in the vertical direction. For example, an 8 bit (grey scale) per pixel, 512x512 pixel digital image.

The digital input image 102 is low-pass filtered in the horizontal direction of the image by a low pass filter 106 and is high pass filtered in the horizontal direction of the image by a high pass filter 108. These horizontally filtered images are then horizontally decimated (down sampled) by a factor of two by decimators 110 and 112 respectively.

5 The decimated low pass filtered image output by the decimator 110 is then low pass filtered in the vertically direction by a low pass filter 114 and is high pass filtered in the vertical direction by a high pass filter 116. The decimated high pass filtered image output by the decimator 112 is low pass filtered in the vertical direction by a low pass filter 118 and is a high pass filtered in the vertical direction by high pass filter 120. The vertically
10 filtered images are then vertically decimated (down sampled) by a factor of two by decimators 122, 124, 126, and 128 respectively. Thus, a LL sub-band 130 of the image 102 is produced by the horizontal low pass filter 106, the horizontal decimator 110, the vertical low pass filter 114, and the vertical decimator 122. Similarly, a HL sub-band 132 is produced by the horizontal low pass filter 106, the horizontal decimator 110, the
15 vertical high pass filter 116, and the vertical decimator 124. Similarly, a LH sub-band 134 is produced by the horizontal high pass filter 108, the horizontal decimator 112, the vertical low pass filter 118, the vertical decimator 126. Lastly, a HH sub-band 136 is produced by the horizontal high pass filter 108, the horizontal decimator 112, the vertical high pass filter 120, and the vertical decimator 128.

20 As shown, the wavelet transformer 100 generates four sub-bands LL 130, HL 132, LH 134, and HH 136, each of the same size $n \times n$ from an input image 102 of size $2n \times 2n$. Each of these sub-bands contains a $\frac{1}{4}$ of the number of pixels as in the original image 102. The LL sub-band is a $\frac{1}{4}$ resolution image of the original image 102. The HL and LH sub-bands contain horizontal and vertical edge information of the original image 102.
25 The HH sub-band contains high frequency information in both the vertical and horizontal direction of the original image 102. Together these sub-bands LL 130, HL 132, LH 134, and HH 136 can be used to restore the original image 102.

The wavelet transformer 100 utilises a two-dimensional separable wavelet transform. Consequently, the 2-dimensional wavelet transforming of the image is
30 separated into two filtering stages, namely a horizontal line filtering stage (106, 108, 110, 112) and a vertical line filtering stage (114, 116, 118, 120, 122, 124, 126, and 128). The horizontal line filtering stage filters and decimates the original image in its horizontal direction and outputs two sub-bands L and H, each of the same size $2n \times n$. The vertical line filtering stage filters and decimates these L and H sub-bands in the vertical direction

and outputs the sub-bands LL 130, HL 132, LH 134, and HH 136, each of the same size $n \times n$.

Fig. 2 shows a schematic overview of a prior art inverse two-dimensional separable wavelet transformer 200 suitable for reconstructing an original image transformed by a wavelet transformer shown in Fig. 1. As shown, the sub-bands LL 130, HL 132, LH 134, and HH 136, are interpolated in the vertical direction (up sampled) by a factor of two by interpolators 202, 204, 206, and 208. The up-sampled sub-bands LL 130, HL 132, are respectively filtered in a vertical direction by a low pass filter 210 and a high pass filter 212 and then added by an adder 218 to form a sub-band L. The up-sampled sub-bands LH 134, HH 136, are respectively filtered in a vertical direction by a low pass filter 214 and a high pass filter 216 and then added by an adder 220 to form a sub-band H. These two sub-bands L and H are interpolated in the horizontal direction (up-sampled) by a factor of two by interpolators 222 and 224. The up-sampled sub-bands L and H are then respectively filtered in the horizontal direction by a low pass filter 226 and a high pass filter 228 and then added together by an adder 230 to produce a reconstruction 232 of the original image 102.

Between the wavelet transformer 100 and the inverse transformer 200, the sub-bands LL, HL, LH, and HH may be quantised and entropy encoded so as to achieve compression. During the decompression stage, these entropy encoded sub-bands may be entropy decoded prior to feeding to the wavelet decoder 200.

The filters 106, 108, 114, 116, 118, 120, 210, 212, 214, 216, 226, and 228 typically are digital filters, such as finite impulse response (FIR filters) which ideally are linear and require a small number of taps, and which perform a convolution of the form:

$$y(n) = \sum_k h(n-k)x(k) \quad \text{Eqn. (1)}$$

A major disadvantage of the above-mentioned wavelet transformers and inverse wavelet transformers is that they resort to large memory stores (not shown) to store intermediate results generated between the horizontal and vertical filters. As mentioned previously, this intermediate data is generated and read in an iterative process and adversely affects the performance.

30 First Arrangement

Fig. 3 shows a schematic representation of part of an inverse two-dimensional separable DWT transformer in accordance with a first arrangement 300. The arrangement 300 comprises a vertical filter 302 feeding a horizontal filter 304 via an adder 306 and an

interpolator 308. The arrangement 300 forms that part of the two-dimensional separable DWT arrangement 200 corresponding to the vertical filter 210, adder 218, interpolator 222 and horizontal filter 226. However, the arrangement 300, as distinct from typical wavelet transformers, has no large-scale intermediate memory. Arrangements similar to
5 arrangement 300 may also form the vertical filter 212-horizontal 226 path, the vertical filter 214-horizontal filter 228 path, and the vertical filter 216-horizontal filter 228 path.

For ease of explanation, the operation of the arrangement of Fig. 3 is described with reference to a wavelet transform sub-band having a plurality of transform coefficients $x_{r,s}$ arranged in a plurality of rows r and columns s . The vertical filter 302 has multiple
10 parallel inputs 301 and one output channel 305. In this particular example, the vertical filter 302 has nine taps and nine parallel inputs 301 but is not intended to be limited thereto. Similarly, the horizontal filter 304 has nine taps but is not intended to be limited thereto.

At any one clock cycle, the vertical filter 302 takes as its input nine transform
15 coefficients $\{ x_{i-4,j}, x_{i-3,j}, x_{i-2,j}, x_{i-1,j}, x_{i,j}, x_{i+1,j}, x_{i+2,j}, x_{i+3,j}, x_{i+4,j} \}$ of the wavelet transform sub-band. The vertical filter 302 then calculates and outputs one intermediate transform coefficient $y_{i,j}$ based on these input transform coefficients. That is, the filter 302 undertakes a one-dimensional inverse transform in the vertical direction of the coefficients $\{ x_{i-4,j}, x_{i-3,j}, x_{i-2,j}, x_{i-1,j}, x_{i,j}, x_{i+1,j}, x_{i+2,j}, x_{i+3,j}, x_{i+4,j} \}$. Thus, the resultant
20 intermediate coefficients $y_{i,j}$ are one-dimensional transform coefficients in the horizontal direction of the image at position i,j .

At the next clock cycle, the vertical filter 302 takes as its input the nine transform coefficients $\{ x_{i-4,j+1}, x_{i-3,j+1}, x_{i-2,j+1}, x_{i-1,j+1}, x_{i,j+1}, x_{i+1,j+1}, x_{i+2,j+1}, x_{i+3,j+1}, x_{i+4,j+1} \}$ of the wavelet transform sub-band and outputs another intermediate transform coefficient $y_{i,j+1}$.
25 At the next clock cycle, the vertical filter 302 takes as its input the nine transform coefficients $\{ x_{i-4,j+2}, x_{i-3,j+2}, x_{i-2,j+2}, x_{i-1,j+2}, x_{i,j+2}, x_{i+1,j+2}, x_{i+2,j+2}, x_{i+3,j+2}, x_{i+4,j+2} \}$ of the wavelet transform sub-band and outputs another intermediate transform coefficient $y_{i,j+2}$. The vertical filter 302 continues in this manner for each clock cycle until the end of the row i . The vertical filter 302 then takes as its input the nine transform coefficients $\{ x_{i-3,0}, x_{i-2,0}, x_{i-1,0}, x_{i,0}, x_{i+1,0}, x_{i+2,0}, x_{i+3,0}, x_{i+4,0}, x_{i+5,0} \}$ and continues in a similar manner as the
30 previous row. As can be seen, the vertical filter 302 receives as input groups of coefficients in sequence. Each group $\{ x_{i-4,j}, x_{i-3,j}, x_{i-2,j}, x_{i-1,j}, x_{i,j}, x_{i+1,j}, x_{i+2,j}, x_{i+3,j}, x_{i+4,j} \}$ comprises nine adjacent coefficients arranged in the vertical direction of the image, four on each vertical side of the centre transform coefficient $x_{i,j}$. The vertical filter 302

receives the groups of coefficients in a "raster" type scan order, viz the centre transform coefficient is sequentially input in raster scan order. Thus each transform coefficient x_{ij} will need to be input into the vertical filter 302 a number of times.

The intermediate transform coefficients y_{ij} , y_{ij+1} and so on are fed to the adder 306
5 in a pipeline manner and added to respective intermediate transform coefficients y'_{ij} , y'_{ij+1} from another vertical line filter (not shown). The added transform coefficients are then interpolated in the horizontal direction (up sampled) by an interpolator 308. The interpolated transform coefficients y''_{ij} are then fed to a horizontal line filter 304.

At any one clock cycle, the horizontal filter 304 takes one interpolated coefficient
10 y''_{ij} as input. The horizontal filter 304 then calculates and outputs 310 one coefficient z_{ij-4} based on the input transform coefficients $\{y''_{ij}, y''_{ij-1}, y''_{ij-2}, y''_{ij-3}, y''_{ij-4}, y''_{ij-5}, y''_{ij-6}, y''_{ij-7}, y''_{ij-8}\}$, which are presently stored in the horizontal filter 304. That is, the filter 304 undertakes a one-dimensional inverse transform in the horizontal direction of the coefficients $\{y''_{ij}, y''_{ij-1}, y''_{ij-2}, y''_{ij-3}, y''_{ij-4}, y''_{ij-5}, y''_{ij-6}, y''_{ij-7}, y''_{ij-8}\}$. Thus, the
15 resultant coefficient z_{ij-4} is an reconstructed image coefficient at position $ij-4$ output by filter 226. The reconstructed image is formed by adding the reconstructed image coefficients output by both filters 226 and 228.

At the next clock cycle, the horizontal filter 304 takes as input the next interpolated coefficient y''_{ij+1} and calculates and outputs 310 the inverse transform coefficient based
20 on the coefficients $\{y''_{ij+1}, y''_{ij}, y''_{ij-1}, y''_{ij-2}, y''_{ij-3}, y''_{ij-4}, y''_{ij-5}, y''_{ij-6}, y''_{ij-7}\}$ presently stored therein. In this way, the horizontal filter acts as a shift register arrangement. The horizontal filter 304 continues in this manner until the end of the row i after which it continues at the next row and so on. As can be seen, the horizontal filter 304 is effectively four data points behind the vertical filter in calculating the inverse
25 transform coefficient. The arrangement is such that the inverse transform coefficients are calculated in raster scan order.

It is preferable that edge mirroring be used to overcome the problems in calculating the inverse transform at the edge of the sub-band. For example, at the first clock cycle, the vertical filter 302 takes as its input nine transform coefficients $\{x_{-4,0}, x_{-3,0}, x_{-2,0}, x_{-1,0},$
30 $x_{0,0}, x_{1,0}, x_{2,0}, x_{3,0}, x_{4,0}\}$ where the first (and last) four samples are mirrored as they are read into the vertical filter 302. Alternatively, the first four samples may be set to zero. As mentioned above, the horizontal filter 304 is effectively four data points behind in calculating the inverse transform coefficient. Thus, the horizontal filter can be clocked to commence at the fifth sample in each row with the first (and last) four samples mirrored.

In this way, the first arrangement 300 provides an arrangement for processing two-dimensional separable convolutional kernels such that the filtered output data can be produced in a linear direction with minimal intermediate memory and at a high speed. The first arrangement comprises a specific example of a more general application of the present invention. The present invention is also applicable to two-dimensional separable convolvers. In such an arrangement, the adder 306 and interpolator 308 may be dispensed with.

Turning now to Fig. 4, there is shown a block diagram of a parallel convolver suitable for use as the vertical filter 302 shown in Fig. 3. The vertical filter 302 comprises a finite memory array 402 for receiving the nine transform coefficients $\{x_{i-4,j}, x_{i-3,j}, x_{i-2,j}, x_{i-1,j}, x_{i,j}, x_{i+1,j}, x_{i+2,j}, x_{i+3,j}, x_{i+4,j}\}$ of the wavelet transform sub-band as input 301. The vertical filter 302 further comprises four adders 404, 406, 408, and 410 for adding together the transform coefficients stored in the memory array 402. In this regard, the adder 404 adds together the transform coefficients $x_{i-4,j}$ and $x_{i+4,j}$, the adder 406 adds together the transform coefficients $x_{i-3,j}$ and $x_{i+3,j}$, the adder 408 adds together the transform coefficients $x_{i-2,j}$ and $x_{i+2,j}$, and the adder 410 adds together the transform coefficients $x_{i-1,j}$ and $x_{i+1,j}$. The vertical filter 302, in addition, comprises five memory stores 412, 414, 416, 418 and 420 for storing the filter coefficients h_8, h_7, h_6, h_5 , and h_4 . The vertical filter also comprises four multipliers 422, 424, 426, and 428 for multiplying the added transform coefficients output by the adders 404, 406, 408, and 410 by the filter coefficients h_8, h_7, h_6 , and h_5 stored in memory stores 412, 414, 416, and 418 respectively. In addition, the vertical filter 302 comprises a further multiplier 430 for multiplying the centre transform coefficient $x_{i,j}$ stored in memory array 402 by the centre filter coefficient h_4 stored in memory 420. The results of the multipliers 422, 424, 426, 428, and 430 are then fed to a summer 432 which adds the results of the multipliers and feeds them to an output channel 305 of the vertical filter 302. In this way, the vertical filter 302 outputs an intermediate transform coefficient $y_{i,j}$ one per clock cycle.

The parallel convolver 302 shown in Fig. 4 takes advantage of the symmetry of the inverse wavelet kernel in that it minimizes the number of multipliers. The convolver 302 adds the mirror image counterpart of a transform coefficient around the centre filter coefficient of the filter before applying the multiplication by the filter. Such may be derived by re-arrangement of the convolution computation:

$$y_{i,j} = h_0 x_{i-c,j} + h_1 x_{i-c+1,j} + \dots + h_{2c} x_{i+c,j} \quad \text{Eqn. (2)}$$

where: $c = (N-1)/2$ and N is odd, (in this particular example, $N = 9$).

As the wavelet is symmetrical then: $h_{c+k} = h_{c-k} \dots (k < c)$ thereby enabling re-arrangement of the convolution computation (Eqn. (2)) to:

$$y_{ij} = h_c x_{1j} + h_{c+1}(x_{i+1j} + x_{i-1j}) + h_{c+2}(x_{i+2j} + x_{i-2j}) + \dots h_{2c}(x_{i+cj} + x_{i-cj}) \quad \text{Eqn(3)}$$

Turning now to Fig. 5, there is shown a block diagram of a sequential convolver suitable for use as the horizontal filter 304 shown in Fig. 3. The structure of the horizontal filter 304 is substantially the same as the vertical filter 302, with the exception of the input arrangement of the filter and the values of the filter coefficients h_i . As mentioned previously, the immediate transform coefficients are input into the horizontal filter 304, one at a time, which acts in a shift register manner. The horizontal filter 304 comprises a finite memory array 502 for receiving the nine immediate transform coefficients $\{ y''_{ij}, y''_{ij-1}, y''_{ij-2}, y''_{ij-3}, y''_{ij-4}, y''_{ij-5}, y''_{ij-6}, y''_{ij-7}, y''_{ij-8} \}$ in a sequential manner. The horizontal filter 302 further comprises four adders 504, 506, 508, and 510 for adding together the transform coefficients stored in the memory array 502. Namely, adder 504 adds together the transform coefficients y''_{ij} and y''_{ij-8} , the adder 506 adds together the transform coefficients y''_{ij-1} and y''_{ij-7} , the adder 508 adds together the transform coefficients y''_{ij-2} and y''_{ij-6} , and the adder 510 adds together the transform coefficients y''_{ij-3} and y''_{ij-5} . The horizontal filter 304, in addition, comprises five memory stores 512, 514, 516, 518 and 520 for storing the filter coefficients h_8, h_7, h_6, h_5 , and h_4 . The horizontal filter 304 also comprises four multipliers 552, 524, 526, and 528 for multiplying the added transform coefficients output by the adders 504, 506, 508, and 510 by the filter coefficients h_8, h_7, h_6 , and h_5 stored in memory stores 512, 514, 516, and 518 respectively. In addition, the horizontal filter 304 comprises a further multiplier 530 for multiplying the centre transform coefficient y''_{i+4j} stored in memory array 502 by the centre filter coefficient h_4 stored in memory 520. The results of the multipliers 522, 524, 526, 528, and 530 are then fed to a summer 532 which adds the results of the multipliers and feeds them to an output channel 310 of the horizontal filter 304. In this way, the horizontal filter 304 outputs an inverse transform coefficient one per clock cycle.

Turning now to Fig. 6, there is shown a block diagram of another sequential convolver suitable for use as the horizontal filter 304 shown in Fig. 3. The horizontal filter 304 comprises a finite memory array 602 for receiving five coefficients $\{ y''_{ij+4}, y''_{ij+3}, y''_{ij+2}, y''_{ij+1}, y''_{ij} \}$ in a sequential manner. The coefficients y''_{ij} are input into the memory array 602, one at a time, which acts in a shift register manner. The vertical filter further comprises five memory stores 622, 624, 626, 628, and 630 for storing the filter coefficients h_8, h_7, h_6, h_5 , and h_4 . The horizontal filter 304, in addition, comprises

five multipliers 612, 614, 616, 618, and 620 for multiplying together the filter coefficients and the transform coefficients. Namely, multiplier 612 multiplies the filter coefficient h_8 and transform coefficient y''_{ij+4} , multiplier 614 multiplies the filter coefficient h_7 and transform coefficient y''_{ij+3} , multiplier 616 multiplies the filter coefficient h_6 and transform coefficient y''_{ij+2} , multiplier 618 multiplies the filter coefficient h_5 and transform coefficient y''_{ij+1} , and multiplier 620 multiplies the filter coefficient h_4 and transform coefficient y''_{ij} . The output of the multipliers 612, 614, 616, and 618 are fed to shift registers 604, 606, 608, and 610 wherein the stored products are shifted by one. The output of the multipliers 612, 614, 616, and 618 together with the output of multiplier 620 are also fed to the summer 632. The summer 632 determines and output 305 the sum of these inputs, namely:

$$z_{ij} = h_4 y''_{ij} + h_5 (y''_{ij+1} + y''_{ij-1}) + h_6 (y''_{ij+2} + y''_{ij-2}) + \dots h_8 (y''_{ij+4} + y''_{ij-4}) \quad \text{Eqn. (4)}$$

In this way, the sequential convolver 600 outputs an inverse transform coefficient one per clock cycle. The horizontal filter 304 shown in Fig. 6 also takes advantage of the symmetry of the inverse wavelet kernel in that it reduces the number of multipliers.

Second Arrangement

Turning now to Fig. 7, there is shown an arrangement of part of an inverse two-dimensional separable DWT transformer in accordance with a second arrangement 700. This DWT transformer arrangement 700 is similar to the arrangement shown in Fig. 3, except that the arrangement 700 has the advantage of accessing multiple lines of the sub-band and output multiple coefficients simultaneously. This advantage is achieved by multiple parallel convolvers (701-1, 701-2) and sequential convolvers (304-1, 304-2) arranged in parallel. Namely, the arrangement 700 comprises multiple vertical filters (701-1, 701-2) with a common finite memory array 702 and corresponding multiple horizontal filters (304-1, 304-2) in parallel.

For ease of explanation, only two parallel vertical and horizontal filter paths are shown, although the arrangement 700 is not intended to be limited thereto. The arrangement 700 can comprise many more parallel vertical and horizontal filter paths. Furthermore, the vertical and horizontal filters filter 302 and 304 both have nine taps but again it is not intended to be limited thereto.

The arrangement 700 comprises two vertical filters 701-1 and 701-2 having a common memory array 702. The memory array 702 has parallel inputs for receiving ten transform coefficients $\{x_{i-5j}, x_{i-4j}, x_{i-3j}, x_{i-2j}, x_{i-1j}, x_{ij}, x_{i+1j}, x_{i+2j}, x_{i+3j}, x_{i+4j}\}$ of the wavelet transform sub-band as input. The nine transform coefficients $\{x_{i-4j}, x_{i-3j}, x_{i-2j}, x_{i-1j}, x_{ij}, x_{i+1j}, x_{i+2j}, x_{i+3j}, x_{i+4j}\}$

$x_{i,j}, x_{i+1,j}, x_{i+2,j}, x_{i+3,j}, x_{i+4,j}$ } are then input to the vertical filter 701-1 and the nine transform coefficients $\{ x_{i-5,j}, x_{i-4,j}, x_{i-3,j}, x_{i-2,j}, x_{i-1,j}, x_{i,j}, x_{i+1,j}, x_{i+2,j}, x_{i+3,j} \}$ are then input to the vertical filter 701-2. Namely, the coefficient $x_{i+4,j}$ is input via 1 to adder 704-1, the coefficient $x_{i+3,j}$ is input via 2 to adders 706-1 and 704-2, the coefficient $x_{i+2,j}$ is input via
5 3 to adders 708-1 and 706-2, and so on. In this way, the vertical filters 701-1 and 701-2 simultaneously calculate the intermediate transform coefficients for the locations i,j and $i-1,j$ respectively.

The vertical filters 701-1 and 701-2 are both substantially the same and only filter 701-1 will be described in detail. The vertical filter 701-1 further comprises four adders
10 707-1, 706-1, 708-1, and 710-1 for adding together the transform coefficients stored in the memory array 702-1. Namely, adder 704-1 adds together the transform coefficients $x_{i-4,j}$ and $x_{i+4,j}$, the adder 706-1 adds together the transform coefficients $x_{i-3,j}$ and $x_{i+3,j}$, the adder 708-1 adds together the transform coefficients $x_{i-2,j}$ and $x_{i+2,j}$, and the adder 710-1 adds together the transform coefficients $x_{i-1,j}$ and $x_{i+1,j}$. The vertical filter 701-1, in
15 addition, comprises five memory stores 712-1, 714-1, 716-1, 718-1 and 720-1 for storing the filter coefficients h_8, h_7, h_6, h_5 , and h_4 . The vertical filter also comprises four multipliers 722-1, 724-1, 726-1, and 728-1 for multiplying the added transform coefficients output by the adders 704-1, 706-1, 708-1, and 710-1 by the filter coefficients h_8, h_7, h_6 , and h_5 stored in memory stores 712-1, 714-1, 716-1, and 718-1 respectively. In
20 addition, the vertical filter 701-1 comprises a further multiplier 730-1 for multiplying the centre transform coefficient $x_{i,j}$ stored in memory array 702-1 by the centre filter coefficient h_4 stored in memory 720-1. The results of the multipliers 722-1, 724-1, 726-1, 728-1, and 730-1 are then fed to a summer 732-1 which adds the results of the multipliers and feeds them to an output channel 705-1 of the vertical filter 701-1. Thus, the vertical
25 filters 701-1 and 701-2 outputs intermediate transform coefficients $y_{i,j}$ and $y_{i-1,j}$ respectively one per clock cycle.

The vertical filters 701-1 and 701-2 are coupled to the horizontal filters 304-1 and 304-2 respectively via adders 740 and 742 and interpolators 744 and 746. The intermediate transform coefficients $y_{i,j}, y_{i-1,j}$ are fed to respective adders 740 and 742 in a
30 pipeline manner and added to intermediate transform coefficients $y'_{i,j}, y'_{i-1,j}$ from other vertical line filters (not shown). The added transform coefficients are then interpolated in the horizontal direction (up sampled) by interpolators 744 and 746.

The interpolated transform coefficients $y''_{i,j}, y''_{i-1,j}$ are then fed to respective horizontal line filters 304-1 and 304-2. The horizontal line filters 304-1 and 304-2 each

comprise a sequential convolver of the type described with reference to Figs. 5 or 6. The horizontal filters 304-1 and 304-2 output the inverse transformed image coefficients in a pipeline manner. The arrangement is such that the inverse transform coefficients are calculated two rows at a time in raster order. This has the advantage that band memory
5 can be released except for the region of overlap.

Third Arrangement

Turning now to Fig. 8, there is shown a flow diagram of a method of transforming a two dimensional data set in a first domain to a two-dimensional data set in a second domain. The two-dimensional data set of the first domain comprises a block of
10 coefficients arranged in rows i , and columns j . Similarly, the two dimensional data set of the second domain comprises a block of coefficients arranged in rows i , and columns j . The transformation is preferably a separable two-dimensional convolution. For example, an inverse two-dimensional separable DWT. In this particular method, the convolution is of the following form:

$$15 \quad y_{i,j} = \sum_{k=-c}^{+c} h_{c+k} x_{i+k,j} \quad \text{Eqn. (5)}$$

and

$$z_{i,j} = \sum_{k=-c}^{+c} h_{c+k} y_{i,j+k} \quad \text{Eqn (6)}$$

where $c = (N-1)/2$ and N is odd.

For the purposes of explanation, the convolution length N is taken as nine
20 coefficients long, but is not intended to be limited thereto. Thus in this example c is equal to four.

The method commences at step 800 where any necessary parameters are initialised. In this step 800, the first two-dimensional data set is also loaded into memory. In addition, a small memory array S for storing intermediate coefficients $y_{i,j}$ is generated.
25 This small memory array S is of a fixed size capable of storing $2c + 1$ (i.e. nine) transform coefficients at any one time. During the processing of the method, the memory array S is adapted to sequentially input only one coefficient at a time. This input coefficient is added to the start of the array and previous coefficients of the array are each shifted once down the array. The previous coefficient at the end of the array is shifted out
30 of the array. Preferably, the array can be implemented as a ring buffer. Alternatively, the array can be implemented as a queue having the properties of a shift register.

After step 800, the method then proceeds to step 802, where counter i represents the row number and counter j represents the column number of the coefficients of the first two-dimensional data set are both initialised to zero.

The method includes a loop 820 where the method steps 804 to 812 are performed
5 for each coefficient of the first data set in a raster scan order. This is achieved by incrementing (816) the counters i and j in raster scan order with every pass of the loop 820.

During step 804, the preferred method reads from the data set stored in memory the coefficient x_{ij} and the four coefficients $\{x_{i-4,j}, x_{i-3,j}, x_{i-2,j}, x_{i-1,j}\}$ and $\{x_{i+1,j}, x_{i+2,j}, x_{i+3,j},$
10 $x_{i+4,j}\}$ in column j on either side of the coefficient x_{ij} .

After the reading of the coefficients $\{x_{i-4,j}, x_{i-3,j}, x_{i-2,j}, x_{i-1,j}, x_{ij}, x_{i+1,j}, x_{i+2,j}, x_{i+3,j},$
 $x_{i+4,j}\}$ in step 804, the method then proceeds to step 806, where these coefficients are vertically convolved in accordance with equation (5) generating an intermediate coefficient y_{ij} . The method then proceeds to step 808 where the coefficient y_{ij} is stored
15 in the small memory array S . The coefficient y_{ij} is added to the start of the array and previous coefficients of the array are each shifted once down the array. The previous coefficient at the bottom of the array is dropped from the array. After the input of coefficient y_{ij} , the array S contains the following coefficients $\{y_{ij}, y_{ij-1}, y_{ij-2}, y_{ij-3}, y_{ij-4},$
 $y_{ij-5}, y_{ij-6}, y_{ij-7}, y_{ij-8}\}$.

After step 808, the method proceeds to step 810 where the stored coefficients
20 $\{y_{ij}, y_{ij-1}, y_{ij-2}, y_{ij-3}, y_{ij-4}, y_{ij-5}, y_{ij-6}, y_{ij-7}, y_{ij-8}\}$ in array S are horizontally convolved in accordance with equation (6), thus generating an resultant coefficient z_{ij-4} . The method then outputs the coefficient z_{ij-4} in step 812. As can be seen, the horizontal convolution step 810 is effectively four data points behind the vertical convolution step 806 in
25 calculating the resultant coefficient z_{ij} . The method thus allows a separable 2-dimensional convolution to be performed in raster scan order with minimal memory.

The method then proceeds to step 814, where a check is made whether the last coefficient z_{ij} of the second data set has been generated. If the decision block 814 returns false, then the method continues to step 804 for another pass of the loop 820. If the
30 decision block returns true, then the method proceeds to step 818 where the method terminates.

It is preferable that edge mirroring be used to overcome the problems in calculating the convolution at the edge of the block of the first data set. For example, during the first loop ($i=0$ and $j=0$), the method during vertical convolution 806 takes as its input nine

transform coefficients $\{x_{-4,0}, x_{-3,j}, x_{-2,0}, x_{-1,0}, x_{0,0}, x_{1,0}, x_{2,0}, x_{3,0}, x_{4,0}\}$ where the first (and last) four samples are mirrored. Alternatively, the first four samples may be set to zero. As mentioned above, the horizontal convolution is effectively four data points behind in calculating the image coefficient. Thus, the horizontal convolution 810 may be set to commence at the fifth sample ($j=4$) in each row with the first (and last) four samples mirrored.

Where the 2-dimensional separable convolution is symmetrical, namely $h_{c+k} = h_{c-k}$ ($k < c$), the convolution may be re-arranged in the form

$$y_{i,j} = h_c x_{i,j} + \sum_{k=1}^{+c} h_{c+k} (x_{i+k,j} + x_{i-k,j}) \quad \text{Eqn. (7)}$$

10 and

$$z_{i,j} = h_c y_{i,j} + \sum_{k=1}^{+c} h_{c+k} (y_{i,j+k} + y_{i,j-k}) \quad \text{Eqn (8)}$$

In this situation, the size of small array and the number of multipliers may be reduced. In this case, the reduced array has a size capable of storing $c + 1$ intermediate coefficients.

15 In a still further arrangement, the method may effectively access and input a predetermined number of rows independently. In this case, the vertical convolution, storage, and horizontal buffer are performed independently for each of the predetermined rows. Thus the method is able perform convolutions in a band like manner.

Fourth Arrangement

20 The method of Fig. 8 is preferably practiced using a conventional general-purpose computer system 900, such as that shown in Fig. 9 wherein the process of Fig. 8 may be implemented as software, such as an application program executing within the computer system 900. In particular, the steps of method of Fig. 8 are effected by software comprising coded instructions that are carried out by the computer. It will be appreciated
25 that such software can be implemented in a variety of programming languages and the coding of such languages may be used to readily implement the teachings of the disclosure contained herein. The software may be divided into two separate parts; one part for carrying out the method of Fig. 8; and another part to manage the user interface between the latter and the user. The software may be stored in a computer readable
30 medium, including the storage devices described below, for example. The software is loaded into the computer from the computer readable medium, and then executed by the

computer. The use of the computer program product in the computer preferably effects an advantageous apparatus in accordance with the method of Fig. 8.

5 The computer system 900 comprises a computer module 901, input devices such as a keyboard 902 and mouse 903, output devices including a printer 915 and a display device 914. A Modulator-Demodulator (Modem) transceiver device 916 is used by the computer module 901 for communicating to and from a communications network 920, for example connectable via a telephone line 921 or other functional medium. The modem 916 can be used to obtain access to the Internet, and other network systems, such as a Local Area Network (LAN) or a Wide Area Network (WAN).

10 The computer module 901 typically includes at least one processor unit 905, a memory unit 906, for example formed from semiconductor random access memory (RAM) and read only memory (ROM), input/output (I/O) interfaces including a video interface 907, and an I/O interface 913 for the keyboard 902 and mouse 903 and optionally a joystick (not illustrated), and an interface 908 for the modem 916. A storage
15 device 909 is provided and typically includes a hard disk drive 910 and a floppy disk drive 911. A magnetic tape drive (not illustrated) may also be used. A CD-ROM drive 912 is typically provided as a non-volatile source of data. The components 905 to 913 of the computer module 901, typically communicate via an interconnected bus 904 and in a manner which results in a conventional mode of operation of the computer system 900
20 known to those in the relevant art. Examples of computers on which the embodiments can be practised include IBM-PC's and compatibles, Sun Sparcstations or alike computer systems evolved therefrom.

Typically, the application program of the preferred embodiment is resident on the hard disk drive 910 and read and controlled in its execution by the processor 905.
25 Intermediate storage of the program and any data fetched from the network 920 may be accomplished using the semiconductor memory 906, possibly in concert with the hard disk drive 910. In some instances, the application program may be supplied to the user encoded on a CD-ROM or floppy disk and read via the corresponding drive 912 or 911, or alternatively may be read by the user from the network 920 via the modem device 916.
30 Still further, the software can also be loaded into the computer system 900 from other computer readable medium including magnetic tape, a ROM or integrated circuit, a magneto-optical disk, a radio or infra-red transmission channel between the computer module 901 and another device, a computer readable card such as a PCMCIA card, and the Internet and Intranets including email transmissions and information recorded on

websites and the like. The foregoing is merely exemplary of relevant computer readable mediums. Other computer readable mediums may be practiced without departing from the scope and spirit of the invention.

Industrial Applicability

5 It is apparent from the above that the embodiment(s) of the invention are applicable to the computer and data processing industries. Particularly, the field of digital data compression and digital image compression.

 The foregoing describes only some embodiments of the present invention, and modifications and/or changes can be made thereto without departing from the scope and
10 spirit of the invention, the embodiment(s) being illustrative and not restrictive.

 In the context of this specification, the word "comprising" means "including principally but not necessarily solely" or "having" or "including" and not "consisting only of". Variations of the word comprising, such as "comprise" and "comprises" have corresponding meanings.

The claims defining the invention are as follows:

1. A method of transforming a two-dimensional data set in a first domain to a two dimensional data set in a second domain, wherein the two-dimensional data set of the first domain comprises coefficients arranged in rows and columns, and the method performs the following steps for each said coefficient in sequence in a first direction:
 - convolving a number of said coefficients that are arranged in a second direction transverse to the first direction to produce a corresponding intermediate transformed coefficient;
 - storing said corresponding intermediate transformed coefficient in a finite memory array, wherein said corresponding intermediate coefficient is stored at a start of the array and previously said stored intermediate coefficients are each shifted once down the array with a previously said stored intermediate coefficient at an end of the array being shifted out of the array; and
 - convolving said intermediate transformed coefficients currently stored in said finite memory array to produce a corresponding transformed coefficient.
2. A method as claimed in claim 1, wherein said first direction comprises a raster scan order.
3. A method as claimed in claim 1 or 2, wherein said step of convolving the number of said coefficients, said storing step, and said step of convolving said intermediate transformed coefficients are performed on said coefficients in a number of rows in parallel in raster scan order.
4. A method as claimed in any one of the preceding claims, wherein said step of convolving said intermediate transformed coefficients utilises a convolution having a convolution kernel length N and said finite memory array is adapted to store said number N of said intermediate transform coefficients.
5. A method as claimed in any one of the preceding claims 1 to 3, wherein said step of convolving said intermediate transformed coefficients utilises a convolution having a convolution kernel length $2c + 1$ and said finite memory array is adapted to store $c + 1$ said intermediate transform coefficients.

6. A method as claimed in any one of the preceding claims, wherein said step of convolving a number of said coefficients comprises inverse wavelet transforming said coefficients in one dimension and said step of convolving said intermediate transformed
5 coefficients comprises inverse wavelet transforming said intermediate transformed coefficients in another dimension.

7. A method as claimed in claim 6, wherein prior to convolving said intermediate transformed coefficients, said intermediate transformed coefficients are added to other
10 said intermediate transformed coefficients and upsampled.

8. A method as claimed in any one of the preceding claims 1 to 7, wherein said finite memory array is implemented as a ring buffer.

15 9. A method as claimed in any one of the preceding claims 1 to 7, wherein said finite memory array is implemented as a shift register.

10. Apparatus for transforming a two-dimensional data set in a first domain to a two dimensional data set in a second domain, wherein the two-dimensional data set of the first
20 domain comprises coefficients arranged in rows and columns, and the apparatus comprises:

a first convolver for convolving a number of said coefficients to produce a corresponding intermediate transformed coefficient;

a finite memory array for storing said corresponding intermediate transformed
25 coefficient, wherein said corresponding intermediate coefficient is stored at a start of the array and previously said stored intermediate coefficients are each shifted once down the array with a previously said stored intermediate coefficient at an end of the array being shifted out of the array; and

a second convolver for convolving said intermediate transformed coefficients
30 currently stored in said finite memory array to produce a corresponding transformed coefficient.

11. Apparatus as claimed in claim 10, wherein said second convolver utilises a convolution having a convolution kernel length N and said finite memory array is adapted to store said number N of said intermediate transform coefficients.
- 5 12. Apparatus as claimed in claim 10, wherein said second convolver utilises a convolution having a convolution kernel length $2c + 1$ and said finite memory array is adapted to store $c + 1$ said intermediate transform coefficients.
- 10 13. Apparatus as claimed in any one of the preceding claims 10 to 12, wherein said first convolver inverse wavelet transforms said coefficients in one dimension and said second convolver inverse wavelet transforms said intermediate transformed coefficients in another dimension.
- 15 14. Apparatus as claimed in any one of the preceding claims 10 to 13, wherein said finite memory array is implemented as a ring buffer.
- 15 15. Apparatus as claimed in any one of the preceding claims 10 to 13, wherein said finite memory array is implemented as a shift register.
- 20 16. A computer readable medium including a computer program for transforming a two-dimensional data set in a first domain to a two dimensional data set in a second domain, wherein the two-dimensional data set of the first domain comprises coefficients arranged in rows and columns, and the computer program comprising:
- 25 code for convolving a number of said coefficients to produce a corresponding intermediate transformed coefficient;
- code for storing said corresponding intermediate transformed coefficient in a finite memory array, wherein said corresponding intermediate coefficient is stored at a start of the array and previously said stored intermediate coefficients are each shifted once down the array with a previously said stored intermediate coefficient at an end of the array being shifted out of the array; and
- 30 code for convolving said intermediate transformed coefficients currently stored in said finite memory array to produce a corresponding transformed coefficient.

17. A computer readable medium as claimed in claim 16, wherein said finite memory array is implemented as a ring buffer.

18. A computer readable medium as claimed in claim 16, wherein said finite memory
5 array is implemented as a shift register.

19. An inverse two-dimensional separable wavelet transformer comprising a plurality of stages, at least one said stage comprising:

10 a first filter for inverse one-dimensional wavelet transforming in a first direction coefficients to produce corresponding intermediate transformed coefficients of a first type;

an adder for adding said corresponding intermediate transformed coefficients of a first type to respective intermediate transformed coefficients of a first type from another stage to produce corresponding intermediate transformed coefficients of a second type;

15 a second filter for inverse one-dimensional wavelet transforming in a second direction transverse to the said first direction said corresponding intermediate coefficients of a second type to produce corresponding transformed coefficients, the second filter comprising:

20 a shift register coupled to the adder for storing said corresponding intermediate transformed coefficients of a second type, and

an arrangement for convolving said intermediate transformed coefficients of the second type currently stored in said shift register to produce a corresponding transformed coefficient.

25 20. A transformer as claimed in claim 19, wherein said transformer further comprises a decimator for decimating said added intermediate transformed coefficients of a first type to produce said intermediate transformed coefficients of a second type.

30 21. A transformer as claimed in claim 19 or 20, wherein said convolution arrangement utilises a convolution having a convolution kernel length N and said finite memory array is adapted to store said number N of said intermediate transform coefficients.

22. A transformer as claimed in claim 19 or 20, wherein said convolution arrangement utilises a convolution having a convolution kernel length $2c + 1$ and said finite memory array is adapted to store $c + 1$ said intermediate transform coefficients.

5 23. A method substantially as described herein with reference to Fig. 8 of the accompanying drawings.

24. Apparatus substantially as described herein with reference to Figs. 3,4 and 5, Figs. 3,4, and 6, Fig. 7 , or Figs. 8 and 9 of the accompanying drawings.

10

25. A computer readable medium including a computer program, the computer program substantially as described herein with reference to Figs. 8 and 9 of the accompanying drawings.

15

DATED this THIRTIETH Day of NOVEMBER 2000

CANON KABUSHIKI KAISHA

Patent Attorneys for the Applicant

SPRUSON&FERGUSON

20

30 11 00 21:03

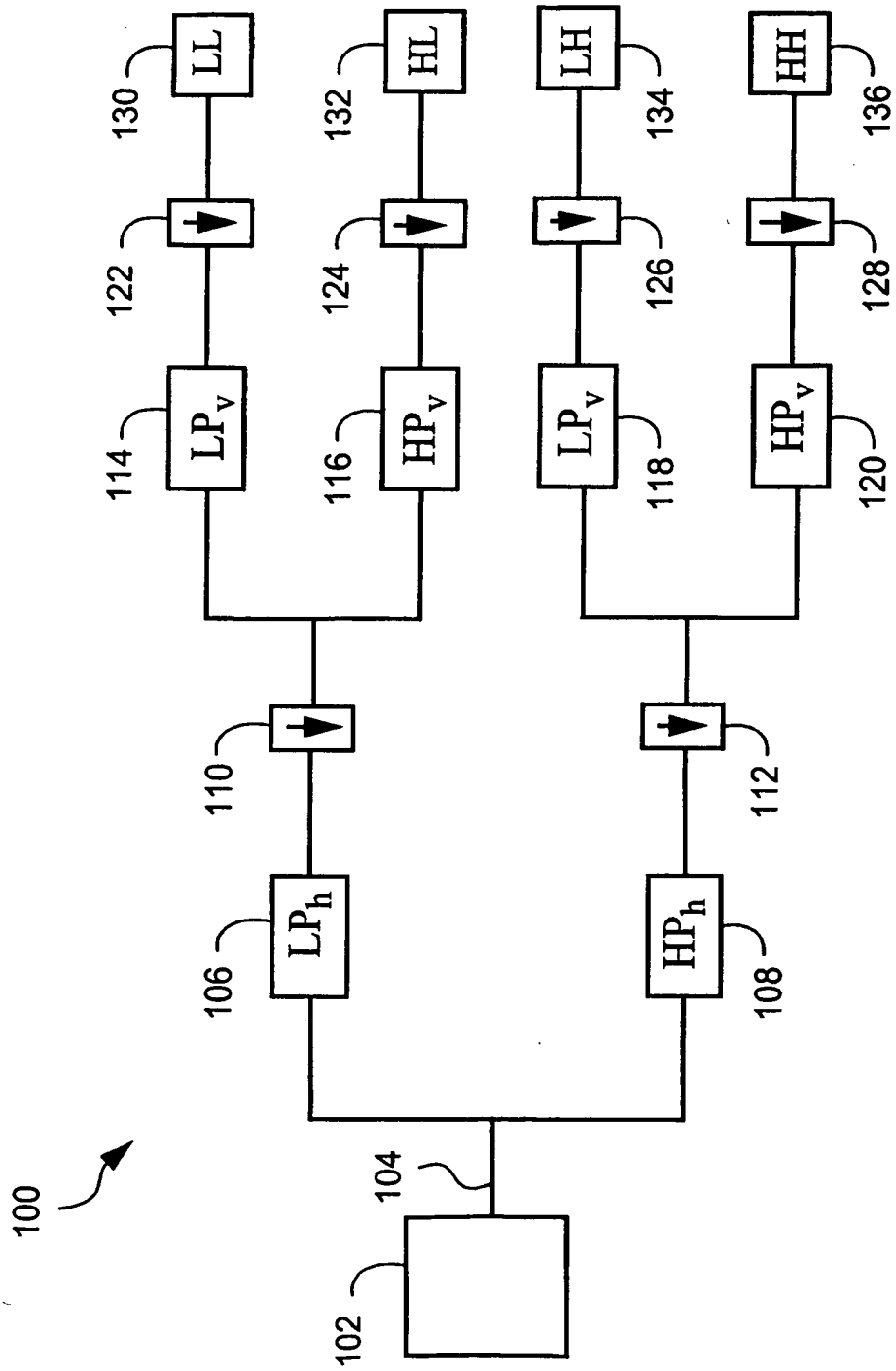


Fig. 1 (PRIOR ART)

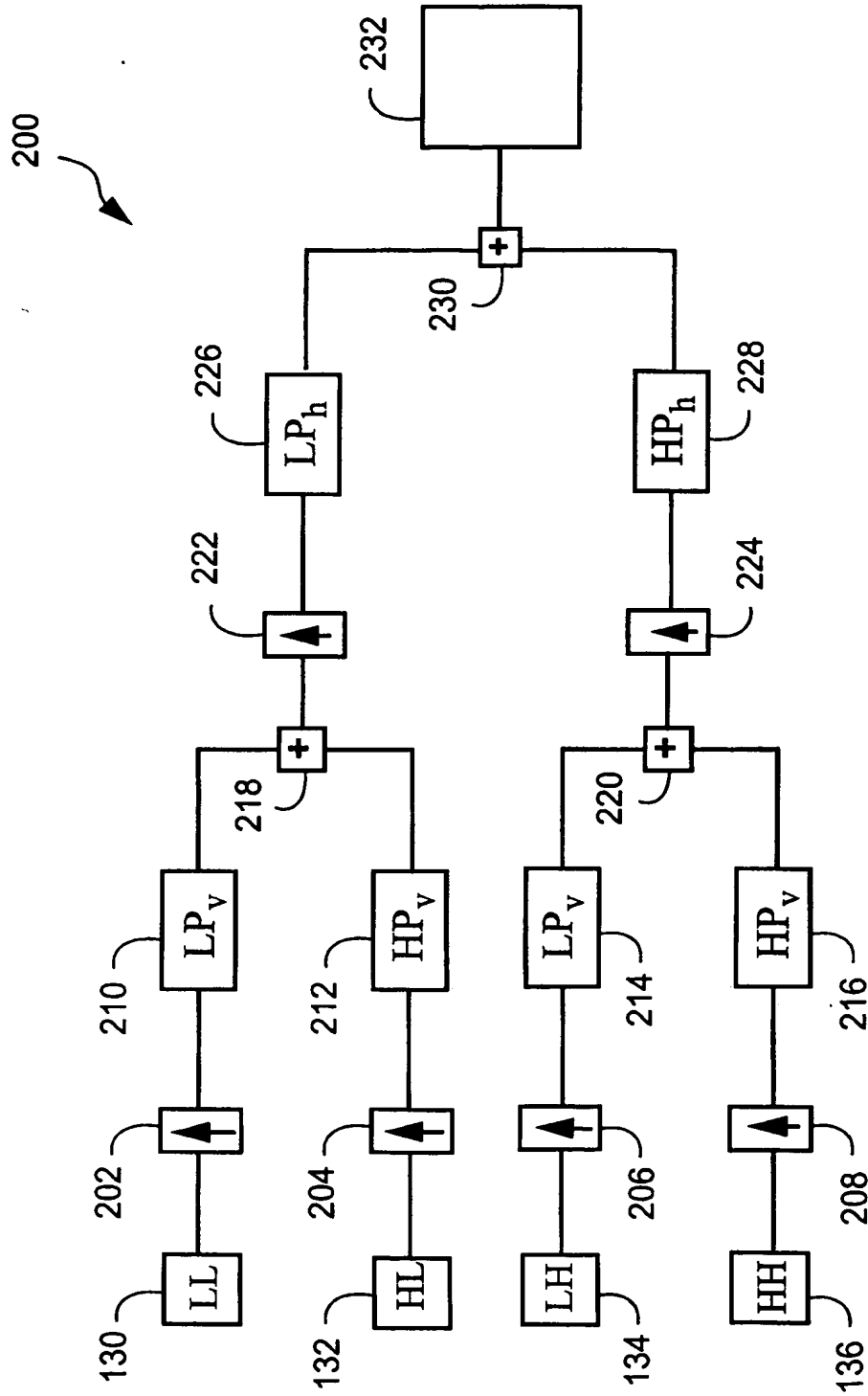
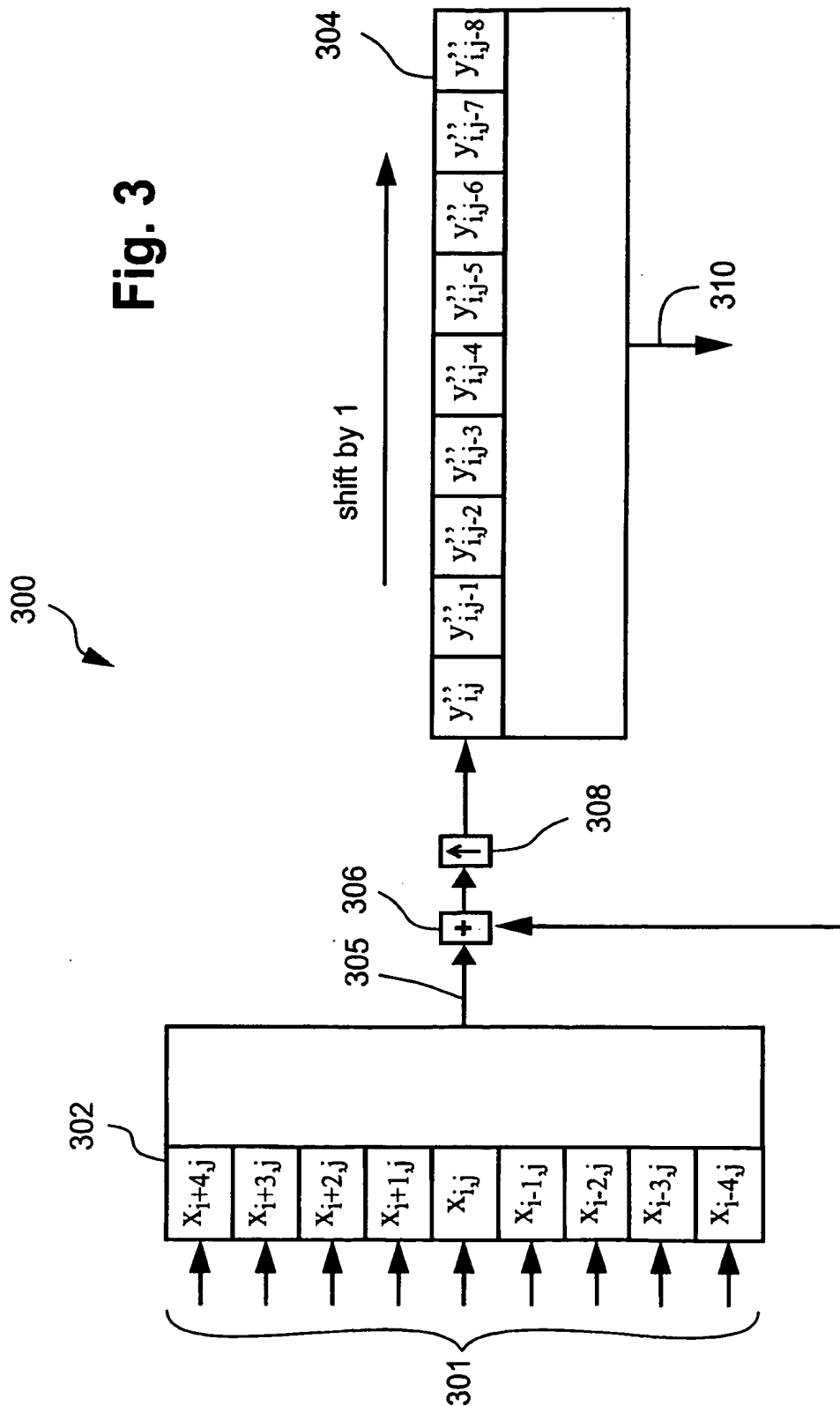


Fig. 2 (PRIOR ART)

Fig. 3



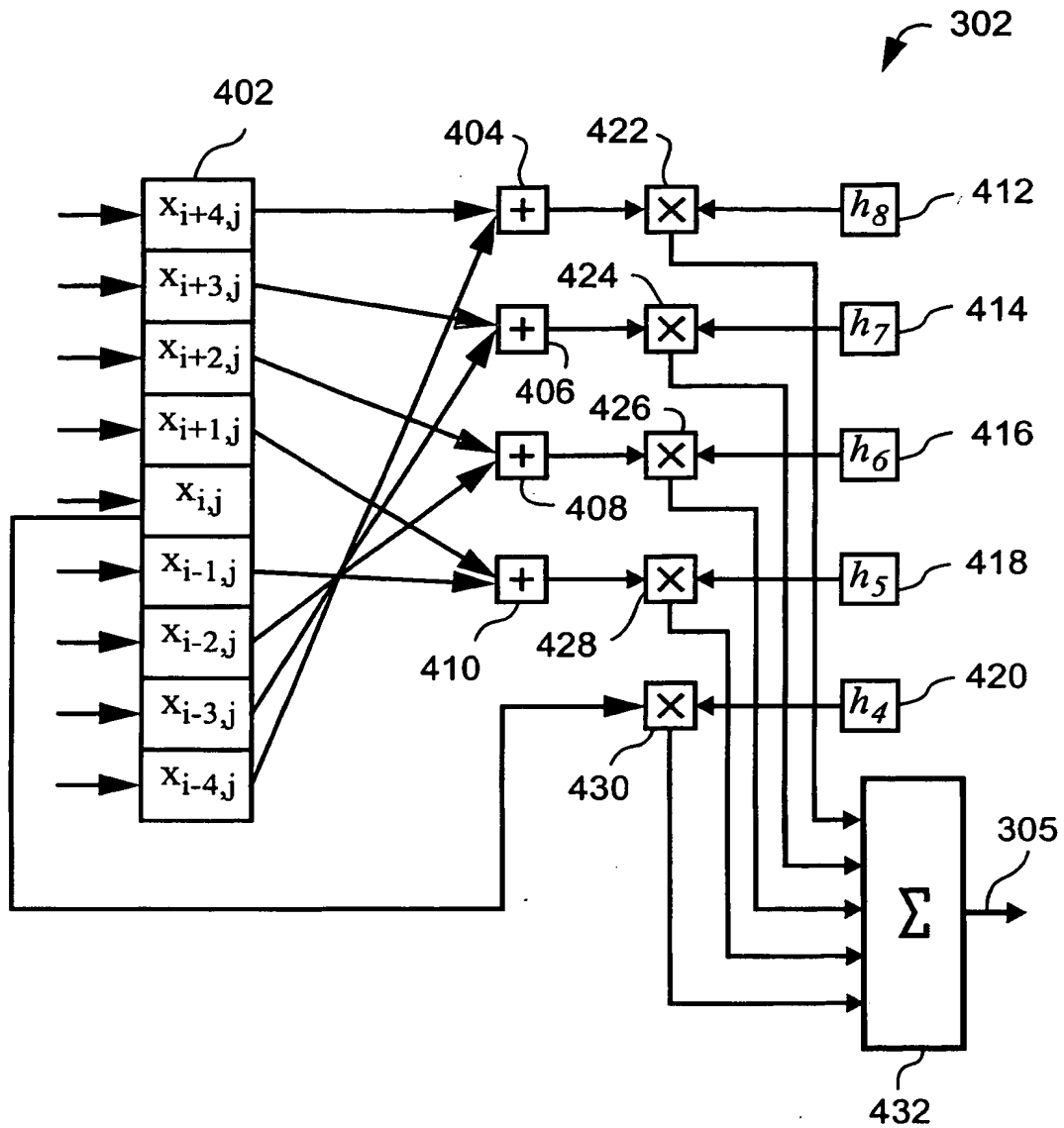


Fig. 4

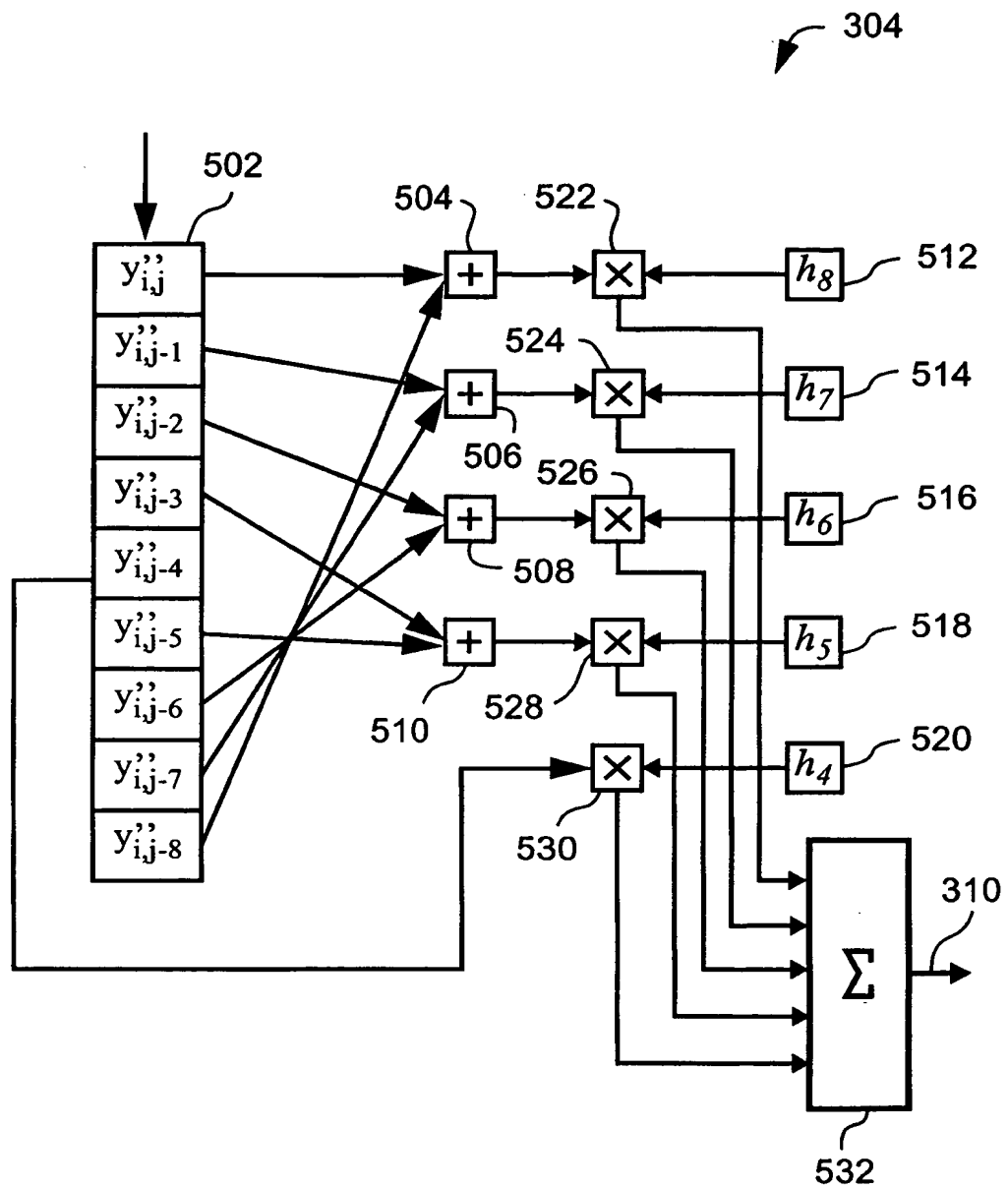


Fig. 5

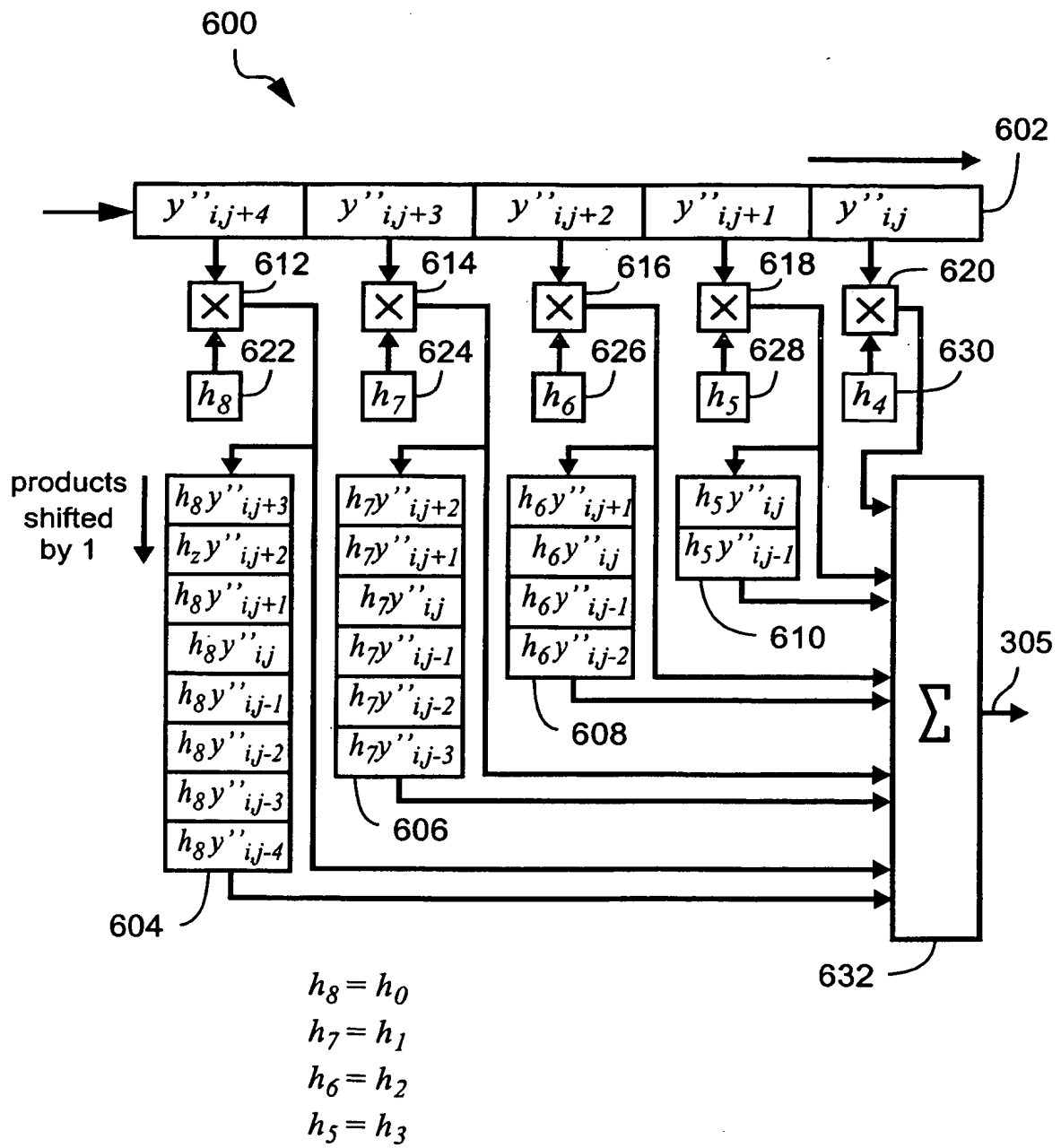


Fig. 6

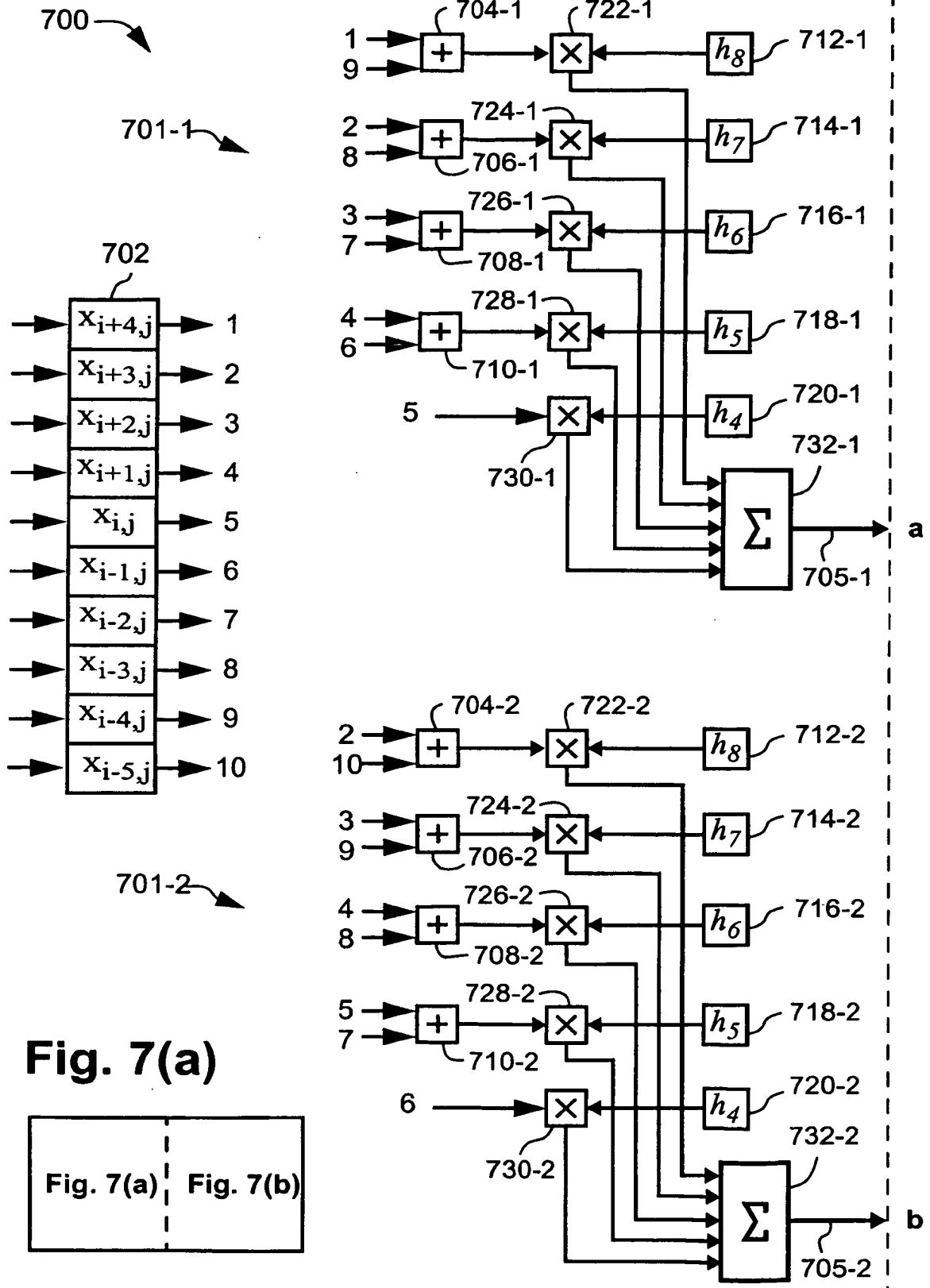
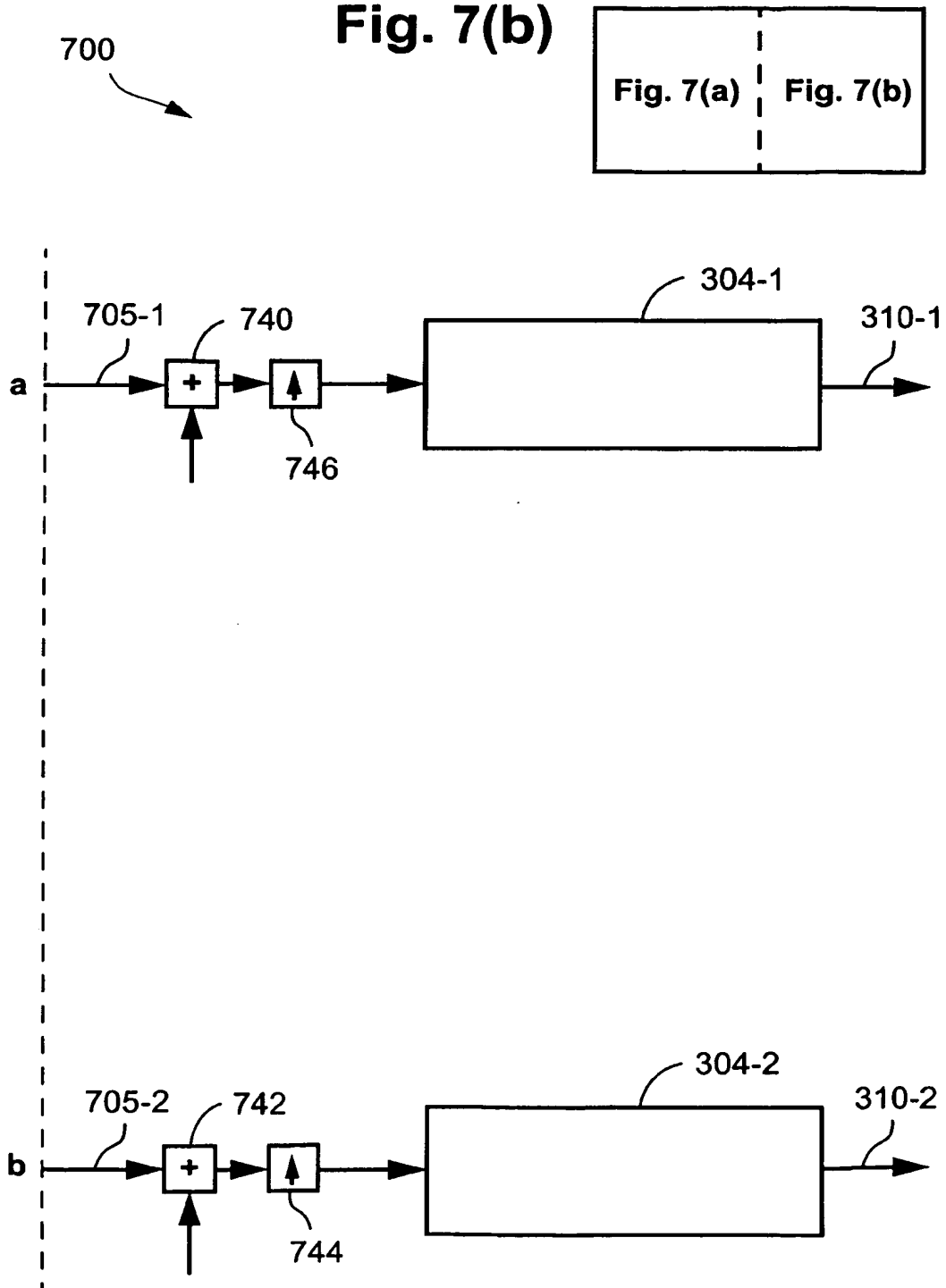
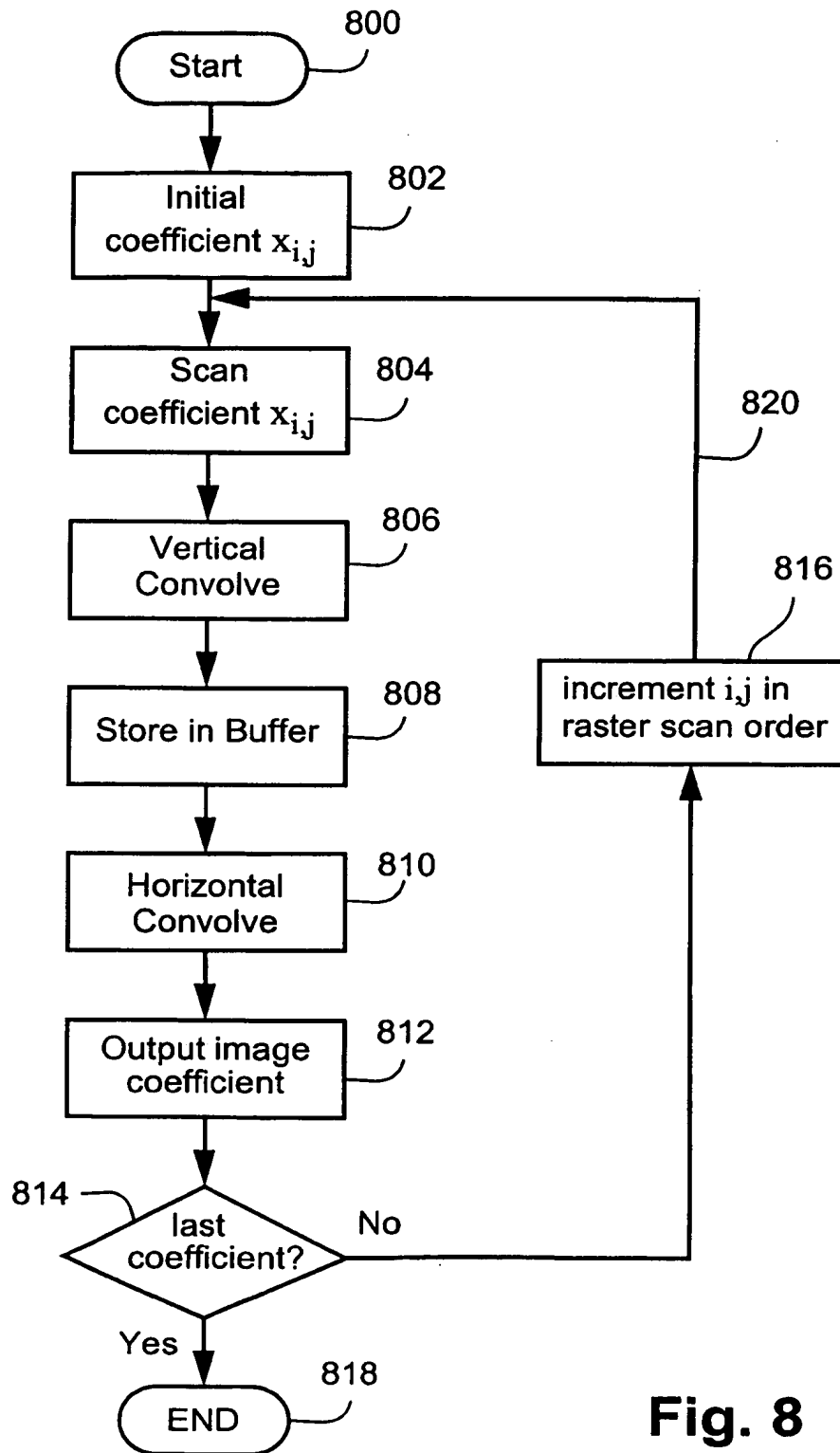
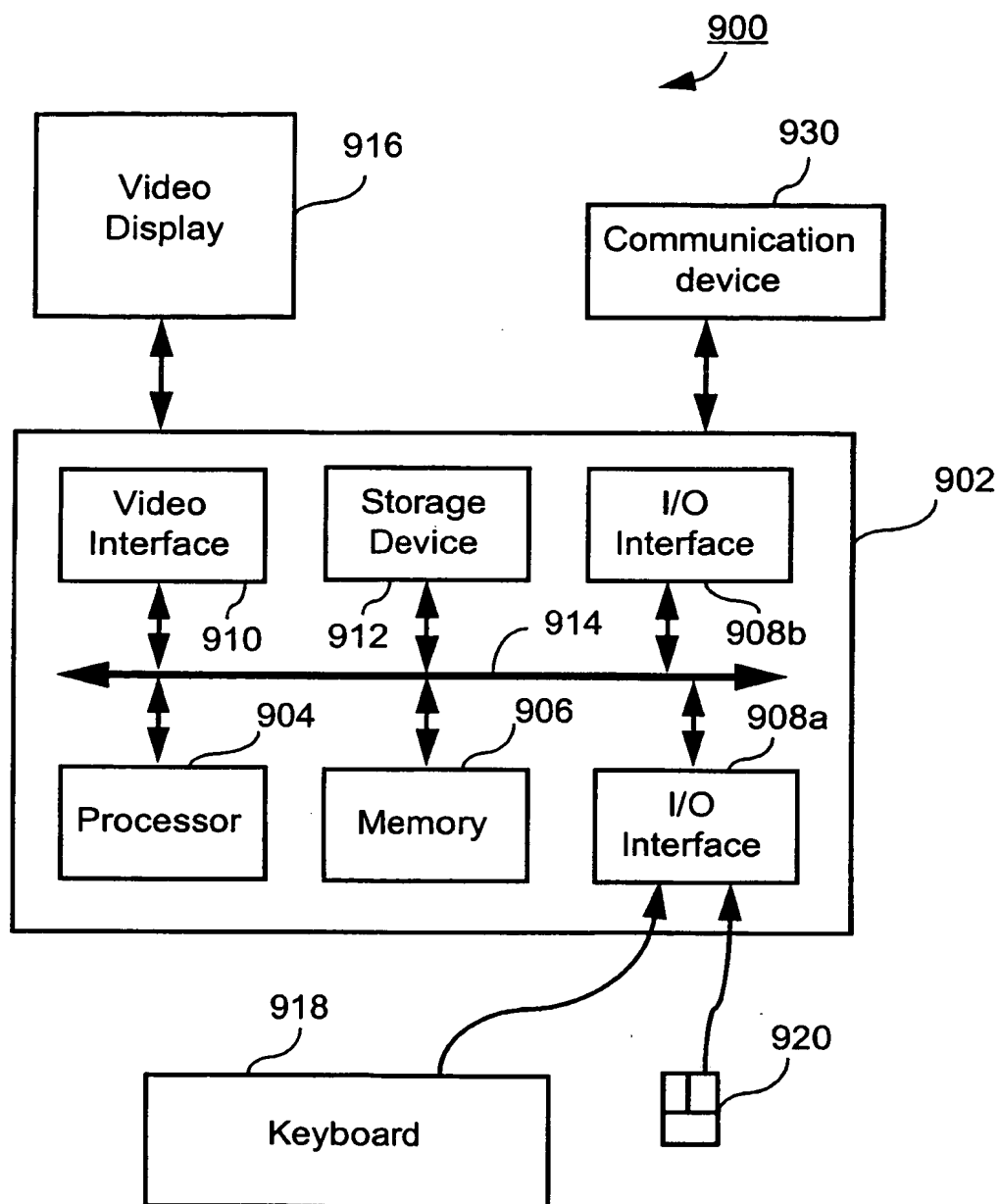


Fig. 7(b)

**Fig. 8**

**Fig. 9**

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.